

# Packages Cdn

**Packages & Framework-less Usage** All Play+ packages are distributed exclusively through your organization's private npm registry . This ensures you are working with secure, vetted, and version-controlled assets aligned with internal governance policies. **Public CDN Not Available** ■ To maintain strict version control and security compliance, Play+ does not offer a public CDN . All assets—including component libraries, stylesheets, and icons—must be installed as npm packages from the configured private registry. This guarantees your applications are built with enterprise-maintained, audited code , supporting traceability and repeatability across environments. **Using Play+ without a Framework** ■ Play+ can be used to build static websites or prototypes without a JavaScript framework. This is especially useful for documentation sites, rapid prototyping, or UI testing. The following guide outlines how to set up Play+ with plain HTML, CSS, and JavaScript . **Quickstart: Static HTML Setup** ■ **1. Create a Project Directory** ■ Initialize a new directory and package.json : `mkdir my-playplus-site cd my-playplus-site npm init -y` **2. Install the Core UI Package** ■ Install the core Play+ UI package from your private registry. This includes precompiled CSS, JavaScript modules, and web components. `npm install @playplus/ui --save` **3. Create the HTML File** ■ Create a basic index.html file in your project root. This file will link to the Play+ assets and contain your markup. **4. Link to Play+ Assets** ■ Add `<link>` and `<script>` tags that point to the required assets from the `node_modules` folder. `<!DOCTYPE html> <html lang = "en"> <head> <meta charset = "UTF-8"> <meta name = "viewport" content = "width=device-width, initial-scale=1.0"> <title> Play+ App </title> <!-- Play+ Core Stylesheet --> <link rel = "stylesheet" href = "node_modules/@playplus/ui/css/playplus.bundle.css"> <!-- Play+ Component Library --> <script type = "module" src = "node_modules/@playplus/ui/dist/playplus.esm.js"> </script> </head> <body> <!-- Example Play+ Component --> <pp-button> Hello Play+ </pp-button> </body> </html>` **Note:** Browsers cannot directly access `node_modules` . To serve this file correctly, copy the required assets into a `public/` folder, or use a static server such as `serve` or `lite-server` . **Using Icons with Play+** ■ Play+ integrates the Lucide icon set—a pixel-perfect, open-source icon system known for its consistency and clarity. Icons are packaged as Web Components via the `@playplus/icons` package, which is automatically installed as a dependency of `@playplus/ui` . You can use icons directly in your HTML with the `<pp-icon>` tag: `<!-- Renders the 'check-circle' icon --> <pp-icon name = "check-circle"> </pp-icon> <!-- Another example --> <pp-icon name = "home"> </pp-icon>` Browse the full list of available icons on the Lucide icon directory . **Summary** ■ Play+ packages are available only via the internal npm registry. Public CDN is intentionally disabled to enforce enterprise security and consistency. Framework-less usage is supported via native Web Components and static asset linking. Lucide-based icons are fully integrated and require no additional installation. For plain HTML setups, remember to resolve `node_modules` assets using a copy or static server workflow.